## NVM Express Technical Errata

| | |
|---|---|
| **Errata ID** | **007** |
| **Affected Spec Ver.** | **NVM Express 1.0** |
| **Corrected Spec Ver.** | |

Submission info

| Name | Company | Date |
|---|---|---|
| Kevin Marks | Dell | 3/29/2011 |

This erratum makes editorial changes to section 3.

Description of the specification technical flaw

***Modify the register fields in section 3.1.1 as shown below:***

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>40:37</td><td>RO</td><td>Impl Spec</td><td>Command Sets Supported (CSS): This field indicates the I/O Command Set(s) ~~command set(s)~~ that the controller supports. A minimum of one command set shall be supported. The field is bit significant. If a bit is set to ‘1’, then the corresponding I/O Command Set ~~command set~~ is supported. If a bit is cleared to ‘0’, then the corresponding I/O Command Set ~~command set~~ is not supported.<br><table><tr><th>Bit</th><th>Definition</th></tr><tr><td>37</td><td>NVM command set</td></tr><tr><td>38</td><td>Reserved</td></tr><tr><td>39</td><td>Reserved</td></tr><tr><td>40</td><td>Reserved</td></tr></table></td></tr>
<tr><td>31:24</td><td>RO</td><td>Impl Spec</td><td>Timeout (TO): This is the worst case time that host software shall wait for the controller to become ready (CSTS.RDY set to ‘1’) after a power-on or reset (refer to section 7.3) ~~event (CC.EN is set to ‘1’ by software)~~. This worst case time may be experienced after an unclean shutdown; typical times are expected to be much shorter. This field is in 500 millisecond units.</td></tr>
<tr><td>18:17</td><td>RO</td><td>Impl Spec</td><td>Arbitration Mechanism Supported (AMS): This field is bit significant and indicates the optional arbitration mechanisms supported by the controller. If a bit is set to ‘1’, then the corresponding arbitration mechanism is supported by the controller. ~~The round robin arbitration mechanism is not listed since all controllers shall support this arbitration mechanism.~~ Refer to section Error! Reference source not found. for arbitration details.<br><table><tr><th>Bit</th><th>Definition</th></tr><tr><td>17</td><td>Weighted Round Robin ~~+~~ with Urgent</td></tr><tr><td>18</td><td>Vendor Specific</td></tr></table><br>The round robin arbitration mechanism is not listed since all controllers shall support this arbitration mechanism.</td></tr>
<tr><td>16</td><td>RO</td><td>Impl Spec</td><td>Contiguous Queues Required (CQR): This field is set to ‘1’ if the controller requires that I/O Submission Queues and I/O Completion Queues are required to be physically contiguous. This field is cleared to ‘0’ if the controller supports I/O Submission Queues and I/O Completion Queues that are not physically contiguous. If this field is set to ‘1’, then the Physically Contiguous bit (CDW11.PC) in the Create I/O Submission Queue and Create I/O Completion Queue commands shall be set to ‘1’.</td></tr>
<tr><td>15:00</td><td>RO</td><td>Impl Spec</td><td>Maximum Queue Entries Supported (MQES): This field indicates the maximum individual queue size that the controller supports. This value applies to each of the I/O Submission Queues and I/O Completion Queues that host software may create. This is a 0’s based value. The minimum value is 1h, indicating two entries.</td></tr>
</table>

***Modify the register fields in section 3.1.5 as shown below:***

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>23:20</td><td>RW</td><td>0</td><td>I/O Completion Queue Entry Size (IOCQES): This field defines the I/O Completion Queue entry size that is used for the selected I/O Command Set. The required and maximum values for this field are specified in the Identify Controller data structure in Figure 65 for each I/O Command Set. The value is in bytes and is specified as a power of two (2^n).</td></tr>
<tr><td>19:16</td><td>RW</td><td>0</td><td>I/O Submission Queue Entry Size (IOSQES): This field defines the I/O Submission Queue entry size that is used for the selected I/O Command Set. The required and maximum values for this field are specified in the Identify Controller data structure in Figure 65 for each I/O Command Set. The value is in bytes and is specified as a power of two (2^n).</td></tr>
<tr><td>15:14</td><td>RW</td><td>0h</td><td>Shutdown Notification (SHN): This field is used to initiate shutdown processing when a shutdown is occurring~~,~~ (i.e., a power down condition is expected.) For a normal shutdown notification, it is expected that the controller is given time to process the shutdown notification. For an abrupt shutdown notification, the host may not wait for shutdown processing to complete before power is lost.<br><br>The shutdown notification values are defined as:<br><br>
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>No notification; no effect</td></tr>
<tr><td>01b</td><td>Normal shutdown notification</td></tr>
<tr><td>10b</td><td>Abrupt shutdown notification</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table>
<br>Shutdown notification should be issued by host software prior to any power down condition and prior to any change of the PCI power management state. It is recommended that shutdown notification also be sent prior to a warm reboot. To determine when shutdown processing is complete, refer to CSTS.SHST. Refer to section Error! Reference source not found. for additional shutdown processing details.</td></tr>
<tr><td>13:11</td><td>RW</td><td>0h</td><td>Arbitration Mechanism Selected (AMS): This field selects the arbitration mechanism to be used. This value shall only be changed when EN is cleared to ‘0’. Host software ~~Software~~ shall only set this field to supported arbitration mechanisms indicated in CAP.AMS. If this field is set to an unsupported value, the behavior is undefined.<br><br>
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Round Robin</td></tr>
<tr><td>001b</td><td>Weighted Round Robin ~~+~~ with Urgent</td></tr>
<tr><td>010b – 110b</td><td>Reserved</td></tr>
<tr><td>111b</td><td>Vendor Specific</td></tr>
</table></td></tr>
</table>

***Modify the register fields in section 3.1.5 as shown below:***

<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>06:04</td><td>RW</td><td>0h</td><td>I/O Command Set Selected (CSS): This field specifies the I/O Command Set ~~command set~~ that is selected for use for the I/O Submission Queues. Host software ~~Software~~ shall only select a supported I/O Command Set ~~command set~~, as indicated in CAP.CSS. This field ~~The command set~~ shall only be changed when the controller is disabled (CC.EN is cleared to ‘0’). The I/O Command Set ~~command set~~ selected shall be used for all I/O Submission Queues.<br><br>
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>NVM Command Set ~~command set~~</td></tr>
<tr><td>001b – 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>00</td><td>RW</td><td>0</td><td>Enable (EN): When set to ‘1’, then the controller shall process commands based on Submission Queue Tail doorbell writes. When cleared to ‘0’, then the controller shall not process commands nor post ~~submit~~ completion queue entries to Completion Queues. When this field transitions from ‘1’ to ‘0’, the controller is reset (referred to as a Controller Reset). The reset deletes all I/O Submission Queues and I/O Completion Queues ~~created~~, resets the Admin Submission Queue and Admin Completion Queue~~s~~, and brings the hardware to an idle state. The reset does not affect PCI Express registers nor the Admin Queue registers (AQA, ASQ, or ACQ). All other controller registers defined in this section are reset. The controller shall ensure that there is no data loss for commands that have ~~been completed~~ had corresponding completion queue entries posted to an I/O Completion Queue prior to the ~~host as part of~~ the reset ~~operation~~. Refer to section 7.3 for reset details.<br><br>
When this field is cleared to ‘0’, the CSTS.RDY bit is cleared to ‘0’ by the controller. When this field is set to ‘1’, the controller sets CSTS.RDY to ‘1’ when it is ready to process commands. The Admin Queue registers (AQA, ASQ, and ACQ) shall only be modified when EN is cleared to ‘0’.</td></tr>
</table>

***Modify the register fields in section 3.1.6 as shown below:***

| Bit | Type | Reset | Description |
|---|---|---|---|
| 03:02 | RO | 0 | **Shutdown Status (SHST):** This field indicates the status of shutdown processing that is initiated by the host setting the CC.SHN field ~~appropriately~~.<br><br>The shutdown status values are defined as:<br><br>**Value** / **Definition**<br>00b – Normal operation (no shutdown has been requested)<br>01b – Shutdown processing occurring<br>10b – Shutdown processing complete<br>11b – Reserved<br><br>To start executing commands on the controller after a shutdown operation (CSTS.SHST set to 10b), a reset (CC.EN cleared to '0') is required. If host software issues commands to the controller without issuing a reset, the behavior is undefined. |
| 01 | RO | 0 | **Controller Fatal Status (CFS):** This field is set to '1' when ~~Indicates that~~ a fatal controller error occurred that could not be communicated in the appropriate Completion Queue. This field is cleared to '0' when a fatal controller error has not occurred. Refer to section 9.5. |
| 00 | RO | 0 | **Ready (RDY):** This field is set to '1' when the controller is ready to process commands after CC.EN is set to '1'. This field shall be cleared to '0' when CC.EN is cleared to '0'. Commands shall not be issued to the controller until this field is set to '1' after the CC.EN bit is set to '1'. Failure to follow this requirement produces undefined results. Host software ~~Software~~ shall wait a minimum of CAP.TO seconds for this field to be set to '1' after setting CC.EN ~~transitions from '0'~~ to '1' from a previous value of '0'. |

***Modify the first paragraph in section 3.1.7 as shown below:***

### 3.1.7 Offset 24h: AQA – Admin Queue Attributes

This register defines the attributes for the Admin Submission Queue and Admin Completion Queue. The Queue ~~ID~~ Identifier for the Admin Submission Queue and Admin Completion Queue is 0h. The Admin Submission Queue's priority is determined by the arbitration mechanism selected, refer to section 4.7. The Admin Submission Queue and Admin Completion Queue ~~Queues~~ are required to be in physically contiguous memory.

***Modify the register fields in section 3.1.8 as shown below:***

| Bit | Type | Reset | Description |
|---|---|---|---|
| 63:12 | RW | Impl Spec | **Admin Submission Queue Base (ASQB):** Indicates the 64-bit physical address for the Admin Submission Queue. This address shall be memory page aligned (based on the value in CC.MPS). All Admin commands, including creation of ~~additional~~ I/O Submission Queues and I/O Completions Queues shall be submitted to this queue. For the definition of Submission Queues, refer to section 4.1. |

***Modify the register fields in section 3.1.9 as shown below:***

| Bit | Type | Reset | Description |
|---|---|---|---|
| 63:12 | RW | Impl Spec | **Admin Completion Queue Base (ACQB):** Indicates the 64-bit physical address for the Admin Completion Queue. This address shall be memory page aligned (based on the value in CC.MPS). All completion queue entries for the commands submitted to the Admin Submission Queue shall be posted to this Completion Queue. This queue is always associated with interrupt vector 0. For the definition of Completion Queues, refer to section 4.1. |

***Modify the register fields in section 3.1.10 as shown below:***

### 3.1.10 Offset (1000h + 8*y): SQyTDBL – Submission Queue y Tail Doorbell

This register defines the doorbell register that updates the Tail entry pointer for Submission Queue y. The value of y is equivalent to the Queue ~~ID~~ Identifier. This indicates to the controller that new commands are ready for processing.

The host should not read the doorbell registers. If a doorbell register is read, the value returned is undefined. Writing a non-existent or unallocated Submission Queue Tail Doorbell has undefined results.

| Bit | Type | Reset | Description |
|---|---|---|---|
| 31:16 | RO | 0 | Reserved |
| 15:00 | RW | 0h | **Submission Queue Tail (SQT):** Indicates the new value of the Submission Queue Tail entry pointer. This value shall overwrite any previous Submission Queue Tail entry pointer value provided. The difference between the last SQT write and the current SQT write indicates the number of commands added to the Submission Queue~~; note that queue rollover needs to be accounted for~~.<br><br>Note: Submission Queue rollover needs to be accounted for. |

***Modify the register fields in section 3.1.11 as shown below:***

### 3.1.11 Offset (1004h + 8*y): CQyHDBL – Completion Queue y Head Doorbell

This register defines the doorbell register that updates the Head entry pointer for Completion Queue y. The value of y is equivalent to the Queue ~~ID~~ Identifier. This indicates Completion Queue entries that have been processed by host software.

The host should not read the doorbell registers. If a doorbell register is read, the value returned is undefined. Writing a non-existent or unallocated Completion Queue Head Doorbell has undefined results.

Host software should ensure it continues to process completion queue entries within Completion Queues regardless of whether there are entries available in a particular or any Submission Queue.

| Bit | Type | Reset | Description |
|---|---|---|---|
| 31:16 | RO | 0 | Reserved |
| 15:00 | RW | 0h | **Completion Queue Head (CQH):** Indicates the new value of the Completion Queue Head entry pointer. This value shall overwrite any previous Completion Queue Head value provided. The difference between the last CQH write and the current CQH entry pointer write indicates the number of entries that are now available for re-use by the controller in the Completion Queue~~; note that queue rollover needs to be accounted for~~.<br><br>Note: Completion Queue rollover needs to be accounted for. |

***Modify the third paragraph in section 3.2 as shown below:***

**Note:** UEFI drivers do not encounter the 1MB limitation, and thus when using UEFI there is not a need for the Index/Data Pair mechanism. Thus, this feature is optional for the controller to support and may be obsoleted as UEFI becomes pervasive.

Disposition log

| | |
|---|---|
| 3/29/2011 | Erratum captured. |
| 5/10/2011 | Erratum ratified. |